

Control de una Plataforma Giroestabilizada en Tiempo Real

Claudio Aciti y Nelson Acosta

INTIA/INCA - Fac. de Ciencias Exactas - Universidad Nacional del Centro de la Prov. de Bs. As.
Paraje Arroyo Seco s/n - Tel/Fax: (02293) 43- 9680 - Tandil (7000) - Buenos Aires - Argentina¹
{caciti, nacosta}@exa.unicen.edu.ar

Resumen

El presente trabajo pretende el desarrollo de un sistema de control en tiempo real de una plataforma giroestabilizada en la cual se ubica una cámara que captura señales de video. El objetivo de la aplicación es que la plataforma mantenga a la cámara apuntando a un lugar fijo, absorbiendo los movimientos del móvil en donde está ubicada, las vibraciones, y cualquier otro tipo de ruido que pueda desestabilizar la señal de video.

1. Introducción

La giroestabilización de plataformas es un tema que gana terreno día a día. Las fuerzas armadas, las fuerzas de seguridad, entre otros, están incorporando a sus vehículos (helicopteros, lanchas, etc.) plataformas giroestabilizadas que les permite apoyar una cámara de video de manera tal que puedan obtener una señal libre de vibraciones y ruidos.

La señal de video obtenida permite tener una visión precisa a gran distancia (puede ser de cientos de metros hasta algunos kilómetros) y brinda la posibilidad de tomar acciones en consecuencia (para realizar búsquedas, persecuciones, controles, etc.). Se pueden usar distintas técnicas para el reconocimiento de patrones (reconocimiento de vehículos por formas, patentes), distinción de colores (en lugares de mucho contraste: agua, tierra, vegetación), seguimiento de objetos (en autopistas, en el mar, en una multitud, etc.), entre otros.

Para controlar una plataforma de este tipo se requiere trabajar a una frecuencia de tiempo real de manera tal que no se pierdan datos de entrada y que al momento de actuar no sea demasiado tarde. Se propone como objetivo la construcción de una plataforma giroestabilizada de cuatro grados de libertad, y una aplicación de control por software usando un sistema operativo de tiempo real.

En la sección 2 se hace una descripción de plataformas giroestabilizadas. En la sección 3 se aborda el tema de sistemas operativos de tiempo real. En la sección 4 se detalla la plataforma realizada y sus componentes. En la sección 5 se presenta el estado actual del proyecto y se plantean líneas futuras. Por último se detalla la bibliografía usada.

2. Plataformas Giroestabilizadas

Por estabilidad se define a la respuesta de un sistema cuando se le mueve de una posición de equilibrio. La estabilidad se divide en dos tipos: estática y dinámica [1][2].

¹ Proyecto financiado por la empresa REDIMEC SRL. Con sede en Tandil, Argentina. <http://www.redimec.com.ar>

- La estabilidad estática se produce cuando un sistema es desplazado de su posición de equilibrio. Se divide en tres tipos: positiva, negativa y neutra. La estabilidad estática positiva genera una fuerza para volver al punto de partida. La neutra permanece en equilibrio en el lugar donde queda. Y la negativa genera una fuerza que lo aleja del punto de equilibrio. En un sistema multidimensional se debe considerar la estabilidad para cada uno de sus ejes por separado
- La estabilidad dinámica se da cuando el movimiento del sistema produce una fuerza que se opone a un movimiento. Es común que suceda que las fuerzas tendientes a recuperar la posición de equilibrio sean tan grandes que fuercen al sistema a ir mas allá de la posición inicial. La estabilidad dinámica es la propiedad que amortigua estas oscilaciones haciéndolas cada vez menores en intensidad.

Las plataformas giroestabilizadas utilizan giróscopos que miden velocidades angulares, en torno a un eje fijo. La posición de la plataforma también es importante ya que puede estar puesta de diferentes formas: colgada, sostenida de costado, apoyada, etc. Todos estos casos hacen variar el problema de la estabilización (Figura 1).

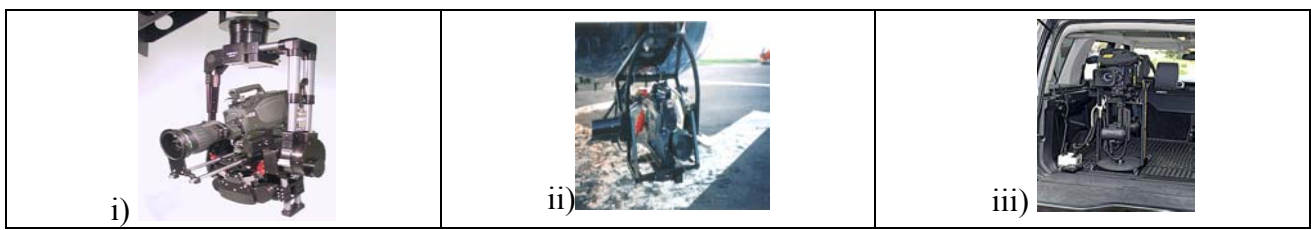


Figura 1. Ejemplos de Plataformas Giroestabilizadas que se utilizan para poner cámaras de video. En i), la plataforma está colgada. En ii) la plataforma está puesta de costado, y en iii) la plataforma está apoyada.

3. Tiempo Real

Se define de Tiempo Real a los sistemas que deben responder ante estímulos generados por el entorno dentro de un período de tiempo finito especificado. En sistemas en donde se trabaja con señales del mundo exterior se debe tener el cuidado de muestrear las señales a una misma frecuencia. En sistemas operativos estándares la velocidad es alta. Pero esta velocidad no es tan importante si no se tiene el control de la frecuencia con la que se muestrea la señal [3] [4] [5] [6].

3.1 Muestreo de Señales

Una señal analógica, compuesta por infinitos valores, al momento de ser convertida al formato digital pierde muchos valores. Si es convertida con muestras que se toman a una frecuencia de tiempo muy pequeña (ns), la pérdida de datos intermedios no se nota. Ahora, si la frecuencia es mayor (ms), la pérdida de datos se hace notar y puede repercutir en el resultado final. Si la señal es muestreada rápidamente pero la frecuencia no es constante, entonces se tiene una señal deformada.

En la Figura 2, se puede apreciar en primer lugar una señal continua. En el gráfico del medio, la misma señal muestreada de forma constante a dos intervalos diferentes, en ambos casos la señal

pierde datos pero no pierde la forma. En el gráfico de abajo, la señal muestreada generalmente a gran rapidez, y por momentos, mas lenta. Se puede ver claramente que la señal original queda deformada.

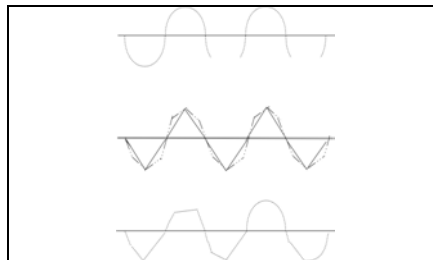


Figura 2. En el primer gráfico se ve la señal original y en los gráficos de abajo, se puede ver la misma señal muestreada de de diferentes formas.

3.2 Tiempo de respuesta de un sistema

El tiempo de respuesta es crítico. Si a un cierto tiempo el sistema no responde, debe desechar el resultado y tomar datos nuevos para que la respuesta sea coherente. En estos casos el uso de un sistema operativo de tiempo real, que si bien no garantiza velocidad, si dá garantías de exactitud en la frecuencia de muestreo.

3.3 Sistemas Operativos de Tiempo Real

Los Sistemas Operativos de Tiempo Real brindan la posibilidad de tener control del tiempo en que se deben ejecutar las tareas. En el mercado actual existe una gran variedad de sistemas operativos de tiempo real. Entre ellos, se destacan: RTLinux, ADEOS, QNX (en todos los casos software propietario), y RTAI (software libre), entre otros. RTAI (Real Time Application Interfaces) es una implementación de Linux para tiempo real basada en RTLinux que añade un pequeño kernel de tiempo real bajo el kernel estándar de linux y trata al kernel de linux como una tarea con la menor prioridad (Figura 3). RTAI además proporciona una amplia selección de mecanismos de comunicación entre procesos y otros servicios de tiempo real. Adicionalmente, RTAI proporciona un módulo llamado LXRT para facilitar el desarrollo de aplicaciones de tiempo real en el espacio de usuario [6] [7] [8] [9] [10].

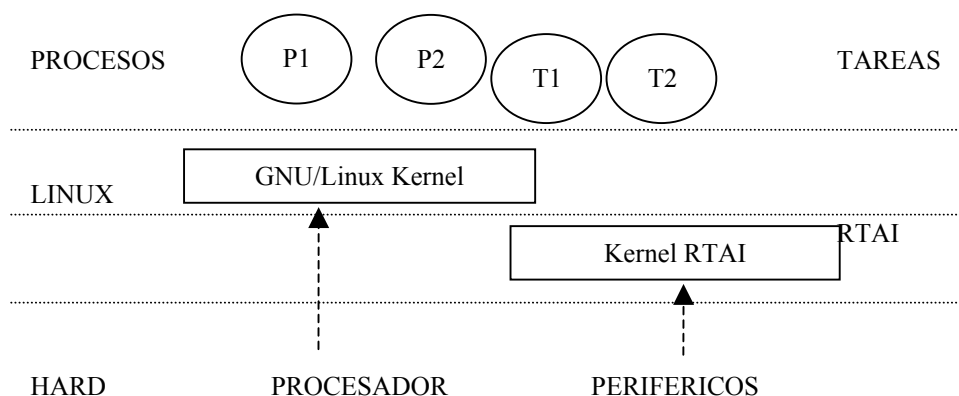


Figura 3: Arquitectura de un sistema operativo GNU/Linux con un parche RTAI.

4. Arquitectura de Tiempo Real Desarrollada

En RTAI las aplicaciones de tiempo real se ejecutan con una prioridad mayor al kernel de linux. De esta forma, una aplicación de alta prioridad no se ve interferidas por otras de menor prioridad. Este parche en el sistema operativo posibilita que si una aplicación que se ejecuta en tiempo real no necesita al procesador, entonces el sistema operativo de oficina GNU/Linux se ejecute normalmente. Para realizar este trabajo se necesitan 3 tareas de tiempo real duro (prioridad máxima). Estas tareas tienen prioridades diferentes y tienen comunicación entre sí.

- La primer tarea (T1) se encarga de leer los datos, cada 1ms, que provienen del DMA (Acceso Directo a Memoria) del conversor y los guarda en buffers. El DMA lee aproximadamente unos 5000 valores por milisegundo de cada canal de entrada. Existe un buffer circular para cada señal. Esta tarea se ejecuta en tiempo real duro con prioridad 2.
- La segunda tarea (T2) toma los datos de los buffers que guardó la tarea anterior. Calcula el promedio de cada buffer, y los envia a la función de control. El controlador se encarga de calcular los resultados y enviarlos a los motores de salida por medio de los conversores de datos. Para realizar el control se pueden utilizar controladores difusos, adaptativos, PIDs, o de cualquier otro tipo. Esta tarea se ejecuta con prioridad 3, es decir que es menos importante que la tarea anterior.
- La tercer tarea (T3) corre con máxima prioridad y está atenta a cualquier interrupción que venga de parte del teclado. Cuando detecta que una tecla fue presionada, pone la variable stop en 1 y termina. Las otras tareas al detectar este cambio, envian valores 0 a los motores para parar la plataforma y terminan inmediatamente. De esta forma se da por terminada la aplicación y no se deja ningun residuo en los canales de salida. Esta tarea corre con prioridad 1 y se ejecuta cada 30 ms.

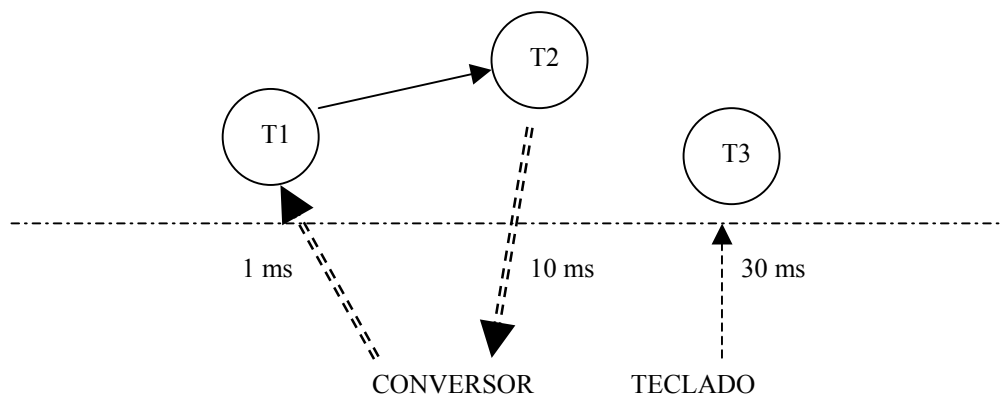


Figura 4. Esquema de la aplicación que corre en tiempo real. Las tareas T1 y T2 se comunican con el conversor analógico / digital, y la tarea T3 lee las interrupciones de teclado.

Un sistema de estas características asegura que las tareas se ejecutan en el tiempo solicitado con una alta precisión. Evitando así latencias indeseadas que lleven a un mal desempeño de la aplicación.

5. Estado Actual del Proyecto

Se construyó con una plataforma de 4 grados de libertad, y se le agregaron los siguientes componentes: a) un giróscopo en azimuth, b) un giróscopo en elevación, c) un potenciómetro para el motor fino en azimuth, d) un potenciómetro para el motor fino en elevación, e) un potenciómetro para el motor grueso en azimuth, y f) un potenciómetro para el motor grueso en elevación. La estabilidad de esta plataforma es controlada por un software y se mueve de un lugar a otro por un joystick que tiene libertad en azimuth y elevación.

Se está trabajando con un sistema operativo en tiempo real: GNU/Linux en una distribución de Debian Sarge con un parche de tiempo real RTAI y un kernel estable backportado 2.6.17-2. El tiempo de muestreo está a un máximo de 49000 por milisegundo aproximadamente, lo que equivale a un poco más de 5000 muestras por canal/ms. Se está utilizando dos conversores analógicos digitales con conexión USB.

Hasta el momento se está utilizando un controlador PID con resultados aceptables. Se prueban diferentes controladores (basados en lógica difusa y redes neuronales) para verificar si los tiempos de cálculo están entre los esperados.

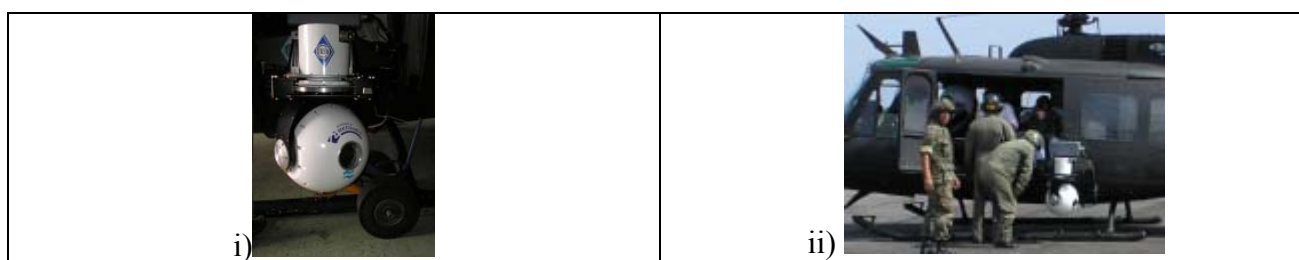


Figura 5. Cámara giroestabilizada desarrollada por REDIMEC SRL en conjunto con CITEFA y probada en la base militar de Campo de Mayo en Diciembre de 2006.

6. Bibliografía

- [1] www.inicia.es/de/vuelo/PBV/PBV16.html
- [2] <http://wings.avkids.com/Libro/Controls/intermediate/stability-01.html>
- [3] <http://www.aero.polimi.it/~rtai/index.html>
- [4] <http://www.linuxdevices.com/articles/AT6605918741.html>
- [5] <http://rtportal.upv.es/>
- [6] <http://www.rtai.org>
- [7] DIAPM RTAI Programming Guide 1.0. <http://www.rtai.org>
- [8] home.gna.org/adeos/
- [9] <https://qnx.com>
- [10] <https://www.rtlinux.org>